

Reusing devices with memory in device independent quantum key distribution

Matthew McKague* Lana Sheridan†
 Centre for Quantum Technologies
 National University of Singapore

September 24, 2012

Abstract

Recently, it has been suggested [3] that device independent approaches to quantum key distribution may be of limited utility, since standard protocols could leak key to the adversary when devices are used repeatedly. We propose a means by which devices with memories could be reused from one run of a quantum key distribution protocol to the next while bounding the leakage to Eve, under the assumption that one run of the protocol could be completed securely using devices with memories.

1 Introduction

Quantum key distribution protocols allow two distant parties who share some small initial key to grow new shared randomness. Proofs of security for these protocols make assumptions about the behaviour of the devices that the two parties, Alice and Bob, use. Device-independent quantum key distribution (DIQKD) [2, 1, 8] is a concept for protocols that make very few assumptions about Alice and Bob's devices for generating their classical measurement outcomes. They should be able to certify that they have a secure key from the statistics of their measurement outcomes alone, however, they still will need to assume that there are no side channels that can signal from their private laboratories to an eavesdropper, Eve. DIQKD protocols have the important advantage that the measurement devices (and

*matthew.mckague@nus.edu.sg

†cqt1ss@nus.edu.sg

the source of quantum states, the control of which we give to Eve) may not operate as intended, but the protocols can still certify whether a generated key is secure.

Here we are interested in removing even more assumptions about the operation of the measurement devices: we allow that they may have an internal memory that can store arbitrary amounts of quantum or classical information, and that they may have been built by the eavesdropper. In this very untrusting model, the question of whether the measurement devices can be reused has been raised [3]. In this paper, we consider that question, but do not consider how DIQKD might be accomplished in a single round of a protocol using an adversarial device with memory.

Our main contribution is to describe an encryption scheme which allows Alice and Bob to exchange data which is determined by the devices across a public channel without leaking information from the devices to Eve. The encryption remains secure even if the devices have complete information about Alice and Bob's shared secret keys (generated in previous rounds of the protocol) and even if the devices have complete control over the message sent. In the context of DIQKD, this allows Alice and Bob to exchange parameter estimation and error correction data without the devices leaking information about previously generated keys to Eve. This is accomplished using locally generated randomness (independent of the devices) and hash functions to generate encryption keys.

The layout of this paper is as follows. In the next section, we specify and motivate the security model we are working in. In section 3, we introduce modifications to a DIQKD protocol. In section 4 we describe a hash function and prove that it has a property we will need. In section 5 we prove that the new protocol is secure over repeated runs and in section 6 look at the scaling. Section 7 gives the asymptotic key rate achieved by these bounds. In section 8 we discuss how protocol aborts need to be managed and touch on the composability implications.

2 The model

Alice and Bob share some private randomness and would like to grow more key from it using a shared quantum state. However, they do not trust their measuring devices or the state; in fact, they assume that Eve has built the devices and distributes the quantum state. Let us assume that it is possible for them to complete a device-independent quantum key distribution (DIQKD) protocol securely in this setting. There is some recent work that

supports this assumption [4, 9, 10]. They successfully grow some new key on which Eve’s knowledge is bounded to be less than ϵ , quantified using standard trace distance metrics [11]. After this, they would like to reuse their devices to grow more key in another round, but the malicious devices are allowed to have a memories. As well, all shared randomness used in the protocol will be taken from the previously generated keys, and hence is also shared with the devices. We would like to know whether Alice and Bob can accomplish this.

We make the standard assumptions of DIQKD. We are working in the limit of long keys for each run of the protocol. We assume that the untrusted devices can be isolated within Alice and Bob’s laboratories, such that they can receive arbitrary quantum signals from Eve, but can signal only to Alice and Bob and not directly to Eve. We also assume that Alice and Bob can both generate trusted randomness locally. Additionally, we assume Alice and Bob can perform classical processing privately from the untrusted measuring devices in their labs.

This model was first introduced in [3], where the authors argue that Alice and Bob cannot grow further key using the same devices and standard protocols. We show how to modify standard DIQKD protocols to eliminate side channels related to Alice and Bob’s public discussion and show that they can still grow new secret key.

3 The new protocol

The modifications we propose are restricted to the classical post-processing portions of the protocol. The goal of the changes is to prevent the device from having a communication channel back to Eve within the protocol itself. (We assume no side channels.)

1. Eve distributes an entangled state ρ_{ABE} to the devices in Alice and Bob’s labs. Alice and Bob supply random (and independent) lists of basis choices to the devices for the series of measurements and the devices output the results.
2. Alice announces her basis selections publicly to Bob. Where they have chosen the same basis, the measurement result bit should be correlated for Alice and Bob and can become part of the key. When they have chosen different bases, they can check for CHSH violation or perform other parameter estimations.

3. Alice must send to Bob a subset of her outcomes of size ℓ . Since the measurement outcomes are fixed by the untrusted devices, these randomly chosen outcomes could hide a message from the device to Eve about previously grown keys, k' . We call this string $m(k, k')$, since it can also depend on our session encryption key k . Therefore she will encrypt it before sending it.
4. Alice generates a random string r of length n and chooses a string k of length n from her store of previously generated keys. She uses a specially chosen 2-universal hash function $r' = f_r(k) = r \cdot k \bmod 2^\ell$ (see section 4 for details), to generate a new string r' of length ℓ , which she bitwise XORs with $m(k, k')$, to form $\bar{m} = m(k, k') \oplus r'$. She sends Bob the result. She also sends him r , publicly.
5. He uses r and k to recover $m(k, k')$. He performs parameter estimation. He sends a similarly encrypted message to Alice containing a flag bit indicating abort or not, and if not, a second encrypted message containing the detected bit error rate Q , the observed parameters, and an appropriate error correction function, along with his parity check bits. Bob pads this communication with randomness, so it is always of fixed length. If they instead will abort, Bob sends the abort flag and random message instead of the error correction information.
6. Using a publicly chosen hash function they reduce Eve's knowledge of the final key below a chosen bound. They discard the session encryption key k used in the protocol.

The intuition behind the modifications is to frustrate the device's ability to hide a message for Eve in m . We assume that the devices know the final keys, since the raw keys are generated by the devices and Eve can send messages to them on the quantum channel. Therefore the device knows k the key string we will use to encrypt m . We must ensure that the device cannot alter the distribution over k of \bar{m} , even conditioned on Eve knowing r . Eve's information on k is upper bounded by ϵ_0 as guaranteed by the single round DIQKD protocol, so by the properties of 2-universal hash functions, the string \bar{m} is Δ -close to uniformly distributed from her perspective.

4 Aside: 2-universal hash functions

We now introduce 2-universal hash functions, in order to show that the particular function we need is a 2-universal hash function.

Definition 1. A 2-universal family of functions \mathcal{F} is a family of functions $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that, when f is drawn uniformly at random from \mathcal{F} , for every $x_1, x_2 \in \mathcal{X}$

$$P(f(x_1) = f(x_2)) = \frac{1}{|\mathcal{Y}|} \quad (1)$$

Consider the family $\mathcal{F} = \{f_a : a \in \{0, 1\}^n\}$, a type of family introduced in [5], given by

$$f_a(x) = a \cdot x \mod 2^\ell \quad (2)$$

where the multiplication is taken in $\text{GF}(2^n)$.¹ We modify this definition to produce the family \mathcal{F}' by introducing an arbitrary function independent of a , $g : \{0, 1\}^n \rightarrow \{0, 1\}^n$, so that

$$f_a(x) = (a \cdot x) \oplus g(x) \mod 2^\ell. \quad (3)$$

To see that this is 2-universal, let $x_1 \neq x_2$ be given. We wish to count the a for which

$$(a \cdot x_1) \oplus g(x_1) = (a \cdot x_2) \oplus g(x_2) \mod 2^\ell. \quad (4)$$

Note that taking the expression modulo 2^ℓ can be seen as taking the ℓ least significant bits of the string, and can be expressed as taking the expression modulo some element in $\text{GF}(2^n)$. Hence we can rewrite this as

$$a \cdot (x_1 \oplus x_2) = g(x_1) \oplus g(x_2) \mod 2^\ell. \quad (5)$$

Since $x_1 \neq x_2$ the expression has solutions. Indeed, since multiplying by an element of $\text{GF}(2^n)$ is just a bijection on $\{0, 1\}^n$, there is one solution for every member of the equivalence class of $g(x_1) \oplus g(x_2) \mod 2^\ell$, of which there are $2^{n-\ell}$ members. Hence the fraction of a that are solutions is $2^{n-\ell}/2^n = 2^{-\ell}$. Thus \mathcal{F}' is 2-universal.

Note that it is equivalent to take $g(x)$ to be an ℓ -bit string instead of n -bit, since the rest of the bits are dropped in final modulus.

We have thus shown that \mathcal{F} has an interesting property: we can add an arbitrary function of x to the family and it is still 2-universal. It is also interesting to note that if g is instead a function of a (but not x) then the function is again still 2-universal by a similar argument.

¹In particular, a and x are bit strings of length n . Choose the standard polynomial representations of these bit strings, multiply them together, return the n bit string corresponding to the product and then take the ℓ least significant bits as the hash output.

5 Proof of main theorem

Here we use lower case letters to denote classical strings (which may be stored in quantum registers) and uppercase letters to denote the registers storing the strings.

Theorem 1. *Let k' be existing previous keys for use in other applications and $\epsilon > 0$ and let Φ be the DIQKD protocol, modified according to section 3. If we have at the start of a run of the new protocol that*

$$\frac{1}{2} \left\| \rho_{K_0 E} - \rho_{K_0}^{U(n_t+a)} \otimes \rho_E \right\|_1 \leq \epsilon_0, \quad (6)$$

where $\rho^{U(n)} = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} |x_A x_B\rangle \langle x_A x_B|$ and f is a particular 2-universal hash function taking two bit string inputs of length n and returning a bit string of length ℓ such that $r' = f(k, r)$ and let $\bar{m} = m \oplus r'$, then after the run of the protocol, giving the string \bar{m} in the register M' to the quantum system E :

$$\begin{aligned} \frac{1}{2} \left\| \Phi(\rho_{KK'ABE}) - \rho_{K_1}^{U(n_t+a')} \otimes \rho_{K'}^{U(a)} \otimes \frac{1}{2^\ell} \sum_{\bar{m}} |\bar{m}\rangle_{M'} \langle \bar{m}| \otimes \rho_{RE} \right\|_1 \\ \leq \epsilon_0 + 3\epsilon_{hash} + \epsilon_{qkd}, \end{aligned} \quad (7)$$

where ϵ_{qkd} is the security bound for one run of the original DIQKD protocol, and $\epsilon_{hash} = \frac{1}{2} \sqrt{2^{2\ell-n}}$ is the error from the hash function in the new protocol. Thus application keys and new key are still almost independent of E .

Suppose before the start of the run of the protocol that Alice and Bob's untrusted devices, which have quantum memory registers A' and B' , have been used to generate some previous key, which we divide into two parts: the session key k to be used for the next run, and the application keys k' , which Alice and Bob will use (or have already used), for example, to send one-time-pad encoded messages, authenticate protocol messages, etc.

We wish to bound the distance of the final state to the ideal uniform tensor product form independently for each message m that the box, A' , might send to Eve. If it is small for all messages, then it is small for any combined strategy. We want:

$$\forall m, \quad \frac{1}{2} \left\| \rho_{K'EM} - \rho_{K'}^{U(a)} \otimes \rho_{EM} \right\|_1 \leq \epsilon_2. \quad (8)$$

We observe that the message, m , is of length ℓ bits. If we give the message to Eve, it cannot contain more than ℓ bits of information on the

session key k , which is $n > \ell$ bits long. Suppose in one round we can grow a key k_0 , secure from Eve. Eve's state with K_0 before the protocol had the property given in equation (9) and notice that $\rho_{K_0}^{U(n_t+a)} = \rho_K^{U(n_t)} \otimes \rho_{K'}^{U(a)}$ if the session key $|k\rangle \in \mathcal{H}_K$ and the applications key $|k'\rangle \in \mathcal{H}_{K'}$ are taken by partitioning $|k_0\rangle \in \mathcal{H}_K \otimes \mathcal{H}_{K'}$, and where $n_t = n + n' + n''$ which is the total amount of encryption key needed for the three messages in the protocol, one from Alice and two from Bob.

Now in the new run of the protocol, which will need to encrypt the classical data output by the devices, Eve will distribute the quantum state to Alice and Bob, so

$$\frac{1}{2} \left\| \rho_{KK'A'B'E} - \rho_K^{U(n_t)} \otimes \rho_{K'}^{U(a)} \otimes \rho_{A'B'E} \right\|_1 \leq \epsilon_0, \quad (9)$$

Then let the operation of the new protocol be the superoperator $\Phi(\cdot)$. Acting on a state that has ideal uniform keys, it will have the following property:

$$\begin{aligned} \frac{1}{2} \left\| \Phi(\rho_K^{U(n_t)} \otimes \rho_{K'}^{U(a)} \otimes \rho_{A'B'E}) \right. \\ \left. - \rho_{K_1}^{U(n_t+a')} \otimes \rho_{K'}^{U(a)} \otimes \frac{1}{2^{(\ell+\ell')}} \sum_{\bar{m}} |\bar{m}\rangle_{M'} \langle \bar{m}| \otimes \sigma_{A'B'ER} \right\|_1 \\ \leq 3\epsilon_{\text{hash}} + \epsilon_{\text{qkd}}. \end{aligned} \quad (10)$$

This is the case because firstly, we assume that the device independent quantum key distribution protocol produces independent keys, at least up to ϵ_{qkd} and because of the Leftover Hash Lemma. The factor of 3 is due to Alice encrypting one message for Bob, and Bob encrypting two messages (the one-bit abort flag and the error correction information) for Alice. First consider Alice's message. Suppose k is a perfect key. Then, we can safely say for all messages, m , of length ℓ :

$$H_{\min}(K|EmK') \geq n - \ell. \quad (11)$$

The Leftover Hash Lemma against quantum side information [12] states that for a 2-universal hash function with output Z , input X , and seed R

$$\Delta(Z|ER) \leq \epsilon + \frac{1}{2} \sqrt{2^{\ell - H_{\min}^{\epsilon}(X|E)}}, \quad (12)$$

where the distance from uniform, Δ , is given by

$$\Delta(A|B)_{\rho} = \min_{\sigma_B} \frac{1}{2} \left\| \rho_{AB} - \omega_A \otimes \sigma_B \right\|_1. \quad (13)$$

Applying this to our hash using a perfect key we get

$$\Delta(R'|EmK'R) \leq \frac{1}{2}\sqrt{2^{2\ell-n}} := \epsilon_{hash}, \quad (14)$$

which gives a bound when $n > 2\ell$.

We can equivalently write this, for all m , fixing m , as:

$$\min_{\sigma_{A'B'ERM}} \frac{1}{2} \left\| \rho_{K'}^{U(n)} \otimes \rho_{R'A'B'MRE} - \rho_{K'}^{U(n)} \otimes \frac{1}{2^\ell} \sum_{r'} |r'\rangle\langle r'|_{R'} \otimes \sigma_{A'B'ERM} \right\|_1 \leq \epsilon_{hash}. \quad (15)$$

The string Eve will intercept on the classical channel between Alice and Bob is the XOR of r' and m : both classical strings, so that the register R' now holds the classical state which is a distribution over strings \bar{m} . Let us relabel register R' as M' . Now make a change of variable $r' \rightarrow r' \oplus m = \bar{m}$. Since r' is uncorrelated to m, E, R, K' , then \bar{m} is also, and runs uniformly over all possible strings since r' did and m is fixed. Tracing out M can only reduce the trace distance so we have:

$$\min_{\sigma_{A'B'ER}} \frac{1}{2} \left\| \rho_{K'}^{U(n)} \otimes \rho_{M'A'B'RE} - \rho_{K'}^{U(n)} \otimes \frac{1}{2^\ell} \sum_{\bar{m}} |\bar{m}\rangle_{M'} \langle \bar{m}| \otimes \sigma_{A'B'ER} \right\|_1 \leq \epsilon_{hash}. \quad (16)$$

We then combine this with the device independent QKD protocol for a single round to get the map $\Phi(\cdot)$ and equation (10), noting that Alice and Bob will both send messages, and assuming the same security level ϵ_{hash} is chosen for each.

Now, we do not have a perfect key initially, instead we have the bound in equation (9). Applying the superoperator Φ to both states in the expression cannot increase the distinguishability of the states, therefore:

$$\frac{1}{2} \left\| \Phi(\rho_{KK'A'B'E}) - \Phi(\rho_K^{U(n_t)} \otimes \rho_{K'}^{U(a)} \otimes \rho_{A'B'E}) \right\|_1 \leq \epsilon_0. \quad (17)$$

By the triangle inequality using equations (17) and (10)

$$\begin{aligned} \frac{1}{2} \left\| \Phi(\rho_{KK'A'B'E}) - \rho_{K_1}^{U(n_t+a')} \otimes \rho_{K'}^{U(a)} \otimes \frac{1}{2^{(\ell+\ell')}} \sum_{\bar{m}} |\bar{m}\rangle_{M'} \langle \bar{m}| \otimes \rho_{RE} \right\|_1 \\ \leq \epsilon_0 + 3\epsilon_{hash} + \epsilon_{qkd}. \end{aligned} \quad (18)$$

So, even after Eve gets the message, her systems are still almost independent of the state held in K' and the new key in K_1 , provided $n > 2\ell$, $n' > 2$, and $n'' > 2\ell''$ for Alice and Bob's hashes.

6 Composing rounds of the new protocol

In the previous section, we saw that reusing untrusted devices in a new round of QKD using the new protocol caused an increase in the security parameter of the new and old keys by $3\epsilon_{\text{hash}} + \epsilon_{\text{qkd}}$. For comparison, if the devices were trusted, and the original DIQKD protocol was used, this parameter would only have grown by ϵ_{qkd} .

Then composing s rounds of successful key growth together, in the worst case the errors can add:

$$\frac{1}{2} \left\| \Phi^{\otimes s}(\rho_{KK'A'B'E}) - \bigotimes_{i=1}^{s-1} \rho_{K_i}^{U(a'_i)} \otimes \rho_{K_s}^{U(n_t+a'_s)} \otimes \rho_{K'}^{U(a)} \otimes \frac{1}{2^{s\ell}} |\bar{m}\rangle_{M'} \langle \bar{m}| \otimes \rho_{RE} \right\| \leq \epsilon_0 + 3s\epsilon_{\text{hash}} + s\epsilon_{\text{qkd}}, \quad (19)$$

so each additional round can add at most $3\epsilon_{\text{hash}} + \epsilon_{\text{qkd}}$ to Eve's information on the previously grown keys.

Notice that if an abort occurs in round i , the new key $\rho_{K_i}^{U(n_t+a'_i)}$ is not obtained for that round and will not appear in that expression. However, Alice and Bob still sent two encrypted messages to each other in that round, in order to learn that their error rate was above threshold. Therefore, they still must add $2\epsilon_{\text{hash}}$ for that round, though not ϵ_{qkd} . This means that the security parameter will grow even on aborted rounds.

In practice, Alice and Bob should choose a maximum tolerated security loss of all of their keys ϵ_{sec} . This will determine the number of rounds they would be able to grow key in. They should agree to this number of rounds when they begin to use their devices, then stop using and securely destroy the devices after that many rounds. They do not wish to leak information to Eve about the number of rounds that have aborted. (See section 8 for further discussion.)

7 Asymptotic secret key rate

The application key rates achievable with this protocol modification will depend on the key rate of the underlying DIQKD protocol used, and n_t the number of bits of the generated key that need to be used as the session keys for Alice and Bob's encrypted messages in the next round, and therefore cannot be used in other applications.

Since we do not know which DIQKD protocol can be used when the devices have memories, we remain agnostic about the exact rate, however,

we can assume it would take a form:

$$r \geq f(S_{\text{obs}}) - H(A|B) \quad (20)$$

for some function f with S_{obs} an observed parameter (*eg.* a Bell-inequality violation) which is what is achieved by current protocols against memoryless devices [7, 6].

In this new protocol, we do not need to remove the amount of communication $H(A|B)$ required for error correction, since this is encrypted. However, we will remove the amount of key required to encrypt the next round's communication. We now consider how much key this requires. From section 5, we have:

$$\epsilon_{\text{hash}} = \frac{1}{2} \sqrt{2^{2\ell-n}} = \frac{1}{2^{(n-2\ell)/2+1}}, \quad (21)$$

Then $n - 2\ell = O(-\log(\epsilon_{\text{hash}}))$, so for a constant security parameter ϵ_{hash} , the key length, n , needs only exceed twice the message length, 2ℓ , by a constant number of bits.

Now we must determine how large the total amount of encrypted information sent between Alice and Bob must be asymptotically. Suppose the sifted key length in one round is N . The parameter estimation message from Alice to Bob must contain the bit values of an $O(\log N)$ -size subset of this string in order to achieve an estimation error approaching zero. As $N \rightarrow \infty$ the fraction of signals this represents goes to zero. Bob must send to Alice his error correction function results, the size of which will depend on the error rate Q . The amount of communication required will be $Nh(Q) + f(\epsilon_{\text{EC}})$ bits, where $h(\cdot)$ is the binary entropy and $f(\epsilon_{\text{EC}})$ is a function of the security parameter for the error correction ($\epsilon_{\text{EC}} < \epsilon_{\text{prev}}$) that does not depend on N . Therefore as $N \rightarrow \infty$ this also is negligible. Finally, Bob's abort flag requires a constant sized key. Then we can see how the asymptotic key rate will change as compared with the original version of the protocol,

$$r \geq f(S_{\text{obs}}) - 2H(A|B). \quad (22)$$

Notice that asymptotically the key rate does not fall as aborts occur, since in an abort, Bob will send the encoded abort flag, but will not encode the $H(A|B)$ bits of error correction information and rather save his key by sending a string output by his random number generator instead. In the finite key regime however, it is clear that aborts will reduce the amount of generated key that can be used in other applications.

8 Aborts

It may happen that on some rounds Alice and Bob must abort the protocol. However, since the devices A' and B' can cause an abort even on a “good” state $\rho_{A'B'E}$, they can use this as a pretext to signal to Eve, as was observed in [3]. Therefore, Alice and Bob must hide aborts when they occur. As explained in section 3, they can do this since they have encrypted the parameter estimation bits and will also encrypt Bob’s signal as to whether or not to abort. If they abort, they pretend to continue the protocol, but instead of exchanging encrypted information to perform error correction, they send random strings. In this round they do not gain any additional key, but also Eve does not learn that they aborted.

Another concern is that it is possible for the boxes to conduct a denial-of-service attack until Alice and Bob run out of key. If this should occur before the number of rounds that Alice and Bob had agreed to use the devices for, this would also constitute a signal to Eve. They must hide this also, so should it occur, Alice and Bob should simulate the remaining rounds of key growth (sending each other random strings) and then destroy the adversarial boxes securely. This is not a foolproof solution however, since in the meantime Alice and Bob may need to communicate privately. Thus at some point they will be forced to re-key and there is no reason to assume Eve will not notice this. Therefore, it is conceivable that she may gain some information from the fact that this has happened and it seems there is no way to completely avoid that, though Alice and Bob could keep a piece of their initial authentication key from before the first round against this eventuality. (This is similar to the case in trusted-device QKD when Eve executes repeated denial-of-service attacks on Alice and Bob until they run out of key.)

It appears that in this model we cannot think about each run of the device independent protocol as a stand-alone element in a universal compossibility scheme, in which it is public information how much key they have at any given time. Alice and Bob certainly do not want to output on each round whether they succeeded or failed in obtaining key. This may lead to additional considerations. For example, the adversary may expect Alice and Bob to send a one-time-pad encoded message at a particular time during the multi-round life of the devices when they do not have key available to devote to the purpose. If this occurs they can still avoid leaking information to the adversary by sending a random string of the appropriate length instead. (However, this does not accomplish the communication task Alice and Bob presumably wished to accomplish.) Note that in this case, Alice

and Bob have to consider their quantum key distribution in the wider setting in which it is employed to avoid leaking information. Nevertheless, when key is generated in the DIQKD scheme, the resulting key is secure under the trace distance definition given in [11].

9 Conclusions and comments on the model

This model of DIQKD gives a lot of power to the eavesdropper, since Eve is allowed to prepare Alice and Bob’s measuring devices. It is more restrictive to Alice and Bob than other models currently used to describe untrusted device scenarios, where their devices may have manufacturing flaws, but are assumed not to be outright malicious. Those models more realistically represent most cryptographic scenarios today, wherein perhaps a user does not understand the cryptography implemented by his web browser, but he downloaded an authenticated copy from a legitimate business. The business may not have correctly implemented the security, and this is what DIQKD would try to protect against, but it also does not benefit from gaining a reputation for selling users’ credit card information to Eve.

However, this less-trusting model is interesting, first, because it provides bounds for what is possible in other more-trusting DI scenarios, and second, because despite its restrictions, QKD can still be performed without much loss of performance. We have introduced a small modification to a DIQKD protocol, that allows untrusted and malicious devices to be used in repeated round of secure key growth. It is interesting to note that the only part of the protocol that required modification was the classical post-processing. That suggests that perhaps existing QKD protocols could be adapted to other new models readily, simply by considering this portion carefully.

There remain some open questions. We note that our bounds are most likely not tight. It seems likely that a better asymptotic key rate can be obtained, but it is also clear that this new protocol will not exactly achieve the rate of the original DIQKD protocol (one that does not worry about reused untrusted devices). There will be some overhead. It would also be nice to fit this type of protocol into a composability framework, although it is not clear how to do that in existing frameworks. Additionally, there may be other modifications that could be made to existing protocols that accomplish this same task more efficiently.

Acknowledgements This work is funded by the Centre for Quantum Technologies, which is funded by the Singapore Ministry of Education and the Singapore National Research Foundation. We thank Marco Tomamichel

for a helpful discussion and Roger Colbeck for his comments about composability.

References

- [1] Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, Stefano Pironio, and Valerio Scarani. Device-independent security of quantum cryptography against collective attacks. *Phys. Rev. Lett.*, 98:230501, 2007.
- [2] Antonio Acín, Nicolas Gisin, and Lluís Masanes. From bell’s theorem to secure quantum key distribution. *Phys. Rev. Lett.*, 97(12):120405, 2006.
- [3] Jonathan Barrett, Roger Colbeck, and Adrian Kent. Prisoners of their own device: Trojan attacks on device-independent quantum cryptography. 2012. arXiv:1201.4407v3.
- [4] Jonathan Barrett, Roger Colbeck, and Adrian Kent. Unconditionally secure device-independent quantum key distribution with only two devices. 2012. arXiv:1209.0435.
- [5] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143, 1979.
- [6] Esther Hänggi and Renato Renner. Device-independent quantum key distribution with commuting measurements. 2010. arXiv:1009.1833v2.
- [7] Lluís Masanes, Stefano Pironio, and Antonio Acín. Secure device-independent quantum key distribution with causally independent measurement devices. *Nat. Commun.*, 2:238, 2011.
- [8] Stefano Pironio, Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, and Valerio Scarani. Device-independent quantum key distribution secure against collective attacks. *New Journal of Physics*, 11:045021, 2009.
- [9] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems via rigidity of chsh games. 2012. arXiv:1209.0449.

- [10] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of chsh games. 2012. arXiv:1209.0448.
- [11] Renato Renner. Security of quantum key distribution. *Int. J. Quant. Inf.*, 6:1, 2008.
- [12] M. Tomamichel, C. Schaffner, A. Smith, and R. Renner. Leftover hashing against quantum side information. *IEEE Trans. Inf. Theory*, 57:5524, 2011.